

Appl. No.: 09/686,402
Filed: October 10, 2000
Amdt. dated 10/25/2004

Amendments to the Claims:

Please cancel Claims 1-29 and 41-72 and amend Claims 30-32 as follows:

Claims 1-29 (Canceled).

30. (Currently Amended). A method according to Claim 29 for managing the display of information output by a computer program on a display terminal comprising:

providing on a computer-readable medium data modules containing text and graphical data used by the computer program to display information on the display terminal;
receiving a command from the computer program to display data on the display terminal;
retrieving data associated with the command from the data modules; and
displaying the data associated with the command on the display terminal, wherein storage of the text and graphical data in the data modules eliminates the requirement that the text and graphical data for displaying information on the display terminal be contained in the computer program.

*ad
cont*
wherein said providing step provides at least one data module that is a language data module including data strings representing language data, wherein each data string is stored in the language data module and designated by a token, and wherein to display a data string, said receiving step receives a token associated with the data string and a command to display the data string from the computer program and based on the token said retrieving step accesses the language data module and retrieves the data string associated with the token, and wherein said displaying step displays the data string on the display terminal.

wherein said providing step further provides a font module stored in a computer-readable medium, wherein the font module contains font data for displaying 256 standard and extended ASCII characters.

wherein said providing step further provides:

a string data area stored on a computer-readable medium that includes data strings representing language data, wherein each character of each data string is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

Appl. No.: 09/686,402
Filed: October 10, 2000
Amdt. dated 10/25/2004

an extended ASCII font data area stored on a computer-readable medium for storing font data related to extended ASCII characters that are not displayable using the extended ASCII character font data stored in the font module ; and

a double byte character font data area stored on a computer-readable medium for storing font data related to at least one double byte character, and

wherein said providing step provides a string data area wherein characters in a data string that are extended ASCII characters and standard ASCII characters, if any, having ASCII codes less than a selected escape code are stored by their ASCII representations in the string data area, while extended ASCII characters and standard ASCII, if any, having ASCII codes at least as great as the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in the string data area.

21. (Currently Amended). A method according to ~~Claim 29~~ for managing the display of information output by a computer program on a display terminal comprising:

providing on a computer-readable medium data modules containing text and graphical data used by the computer program to display information on the display terminal;

receiving a command from the computer program to display data on the display terminal;

retrieving data associated with the command from the data modules; and

displaying the data associated with the command on the display terminal, wherein storage of the text and graphical data in the data modules eliminates the requirement that the text and graphical data for displaying information on the display terminal be contained in the computer program,

wherein said providing step provides at least one data module that is a language data module including data strings representing language data, wherein each data string is stored in the language data module and designated by a token, and wherein to display a data string, said receiving step receives a token associated with the data string and a command to display the data string from the computer program and based on the token said retrieving step accesses the

Appl. No.: 09/686,402
Filed: October 10, 2000
Amdt. dated 10/25/2004

language data module and retrieves the data string associated with the token, and wherein said displaying step displays the data string on the display terminal.

wherein said providing step further provides a font module stored in a computer-readable medium, wherein the font module contains font data for displaying 256 standard and extended ASCII characters.

wherein said providing step further provides:

a string data area stored on a computer-readable medium that includes data strings representing language data, wherein each character of each data string is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

an extended ASCII font data area stored on a computer-readable medium for storing font data related to extended ASCII characters that are not displayable using the extended ASCII character font data stored in the font module ; and

a double byte character font data area stored on a computer-readable medium for storing font data related to at least one double byte character, and

wherein said providing step provides a string data area wherein characters in a data string that are standard ASCII characters and extended ASCII characters having ASCII codes less than a selected escape code are stored by their ASCII representations in the string data area, while extended ASCII characters having ASCII codes at least as great as the selected escape code and ASCII characters that identify the start of 16 bit double byte characters are encoded into 16 bit values and the encoded values are stored in the string data area.

32. (Currently Amended). A method according to Claim 29 for managing the display of information output by a computer program on a display terminal comprising:

providing on a computer-readable medium data modules containing text and graphical data used by the computer program to display information on the display terminal;

receiving a command from the computer program to display data on the display terminal;

retrieving data associated with the command from the data modules; and

displaying the data associated with the command on the display terminal, wherein storage of the text and graphical data in the data modules eliminates the requirement that the text and

Appl. No.: 09/686,402
Filed: October 10, 2000
Amdt. dated 10/25/2004

graphical data for displaying information on the display terminal be contained in the computer program.

wherein said providing step provides at least one data module that is a language data module including data strings representing language data, wherein each data string is stored in the language data module and designated by a token, and wherein to display a data string, said receiving step receives a token associated with the data string and a command to display the data string from the computer program and based on the token said retrieving step accesses the language data module and retrieves the data string associated with the token, and wherein said displaying step displays the data string on the display terminal.

wherein said providing step further provides a font module stored in a computer-readable medium, wherein the font module contains font data for displaying 256 standard and extended ASCII characters.

wherein said providing step further provides:

a string data area stored on a computer-readable medium that includes data strings representing language data, wherein each character of each data string is a character selected from the group consisting of standard ASCII, extended ASCII, and double byte characters;

an extended ASCII font data area stored on a computer-readable medium for storing font data related to extended ASCII characters that are not displayable using the extended ASCII character font data stored in the font module; and

a double byte character font data area stored on a computer-readable medium for storing font data related to at least one double byte character, and

wherein said providing step provides a string data area including double byte characters that are sequentially encoded and the encoded values representing the double byte characters are stored in the string data area, wherein font data associated with the double byte characters is stored in the double byte character font data area, and wherein at least one extended ASCII character is encoded as a 16 bit value in the string data area with an escape code preceding the ASCII representation of the extended ASCII character, and wherein if the extended ASCII character is not displayable with the extended ASCII character font data stored in said font module, data for the extended ASCII character is stored in the extended ASCII font data area.

Appl. No.: 09/686,402
Filed: October 10, 2000
Amdt. dated 10/25/2004

⁴33. (Original) A method according to Claim ³32, wherein said providing step provides a string data area including double byte characters that are sequentially encoded such that the first double byte character is represented by a two-byte code having a first byte that is one value greater than the escape code and a second byte equal to zero, and wherein remaining unique double byte characters are encoded with sequential 16 bit code values.

⁵34. (Original) A method according to Claim ⁴33, wherein the escape code is selected as E0 hexadecimal, wherein said providing step provides a string data area including double byte characters that are sequentially encoded in the string data area such that the first double byte character is encoded as E100 hexadecimal and the remaining unique double byte characters are encoded with sequential 16 bit values from E101 to FFFF hexadecimal, and wherein the extended ASCII characters having ASCII codes at least as great the selected escape code are encoded in the string data area as a 16 bit code with the selected escape code as the first byte and the code for the extended ASCII character as the second byte.

⁶35. (Original) A method according to Claim ⁵34, wherein to display a data string, said receiving step receives the token associated with the data string and a command to display the data string from the computer program, wherein said retrieving step accesses the location in the string data area where the data string is located, and wherein said display step sequentially displays characters of the data string on the display terminal.

⁷36. (Original) A method according to Claim ⁶35, wherein if a character in the data string is less than 80 hexadecimal, then the character is a standard ASCII character and said retrieving step retrieves the ASCII character code stored in the string data area and said displaying step displays the ASCII character on the display terminal using font data from the font data module.

⁸37. (Original) A method according to Claim ⁷36, wherein if a character in the data string is at least as great as 80 hexadecimal but less than the escape code, then the character is an extended ASCII character having an ASCII code less than the escape code, wherein said retrieving step retrieves the extended ASCII character code stored in the string data area, wherein if the ASCII character is displayable with font data stored in the font data module, said

Appl. No.: 09/686,402
Filed: October 10, 2000
Amdt. dated 10/25/2004

displaying step displays the ASCII character on the display terminal using font data from said font data module, and wherein if the ASCII character is not displayable with font data stored in the font data module, said displaying step displays the ASCII character on the display terminal using font data from the extended ASCII font data area in the language module.

⁹38. (Original) A method according to Claim ⁶35, wherein if a character in the data string is equal to the selected escape code of E0 hexadecimal, then the next character in the data string is an extended ASCII character, wherein said retrieving step retrieves the character code stored in the next byte of the string data area, and wherein said displaying step uses the character code to display the extended ASCII character on the display terminal.

¹⁰39. (Original) A method according to Claim ⁶35, wherein if a character in the data string is greater than the escape code, then the character combined with the next consecutive character in the data string represents a 16 bit double byte character.

*as
cancel*
¹¹40. (Original) A method according to Claim ¹⁰39, wherein the escape code is E0 hexadecimal and the first encoded double byte character is E100 hexadecimal, wherein to display a double byte character said retrieving step subtracts E100 hexadecimal from the character and multiplies the character by a size value representing the pixel display size of a double byte character, wherein the calculated value represents the location offset of the double byte character font data for the character stored in the double byte character font data area, and wherein said retrieving step uses the calculated value to locate the double byte character font data stored in the double byte character font area, and wherein said displaying step displays the double byte character on the display terminal.

Claims 41-72. (Canceled).